

# EERAD3: Event shapes and jet rates in electron-positron annihilation at order $\alpha_s^3$

---

**A. Gehrmann–De Ridder**

*Institute for Theoretical Physics, ETH, CH-8093 Zürich, Switzerland*

*E-mail:* `gehra@phys.ethz.ch`

**T. Gehrmann**

*Physik-Institut, Universität Zürich, Winterthurerstrasse 190,*

*CH-8057 Zürich, Switzerland*

*E-mail:* `thomas.gehrmann@uzh.ch`

**E.W.N. Glover**

*Institute for Particle Physics Phenomenology, Department of Physics,*

*University of Durham, Durham, DH1 3LE, UK*

*E-mail:* `e.w.n.glover@durham.ac.uk`

**G. Heinrich**

*Max-Planck-Institut für Physik, Föhringer Ring 6, D-80805 München, Germany*

*E-mail:* `gudrun@mpp.mpg.de`

**ABSTRACT:** The program **EERAD3** computes the parton-level QCD contributions to event shapes and jet rates in electron-positron annihilation through to order  $\alpha_s^3$ . For three-jet production and related observables, this corresponds to next-to-next-to-leading order corrections, and allows for precision QCD studies. We describe the program and its usage in detail.

**KEYWORDS:** QCD, Jets, LEP and ILC Physics, NLO and NNLO Computations.

---

## Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Event shapes and jet cross sections in perturbative QCD</b>	<b>2</b>
<b>3. Structure of the program EERAD3</b>	<b>4</b>
3.1 Main program	5
3.2 Cross section	5
3.3 Phase space integration	6
3.4 Event shapes and jet cross sections	6
3.5 Calculation of moments of event shape observables	7
3.6 Booking of results into histograms	7
<b>4. Usage of EERAD3</b>	<b>9</b>
4.1 Main program <code>eerad3</code>	9
4.2 <code>eerad3_combine</code>	11
4.3 <code>eerad3_dist</code>	12
<b>5. Summary</b>	<b>14</b>

---

## 1. Introduction

For hadron production at colliders, inclusive quantities like total cross sections correspond to a very simple final state definition and can be computed analytically to very high order in perturbation theory. Experimental measurements are often applying specific selection or reconstruction criteria to the final state, and are therefore less inclusive. Typical exclusive observables are jet rates or event shape distributions. To compute such observables in perturbation theory, it is necessary to implement the definition of the observable at parton-level in the theoretical calculation. Such a parton-level event generator includes all partonic processes relevant up to the required perturbative order. Each individual parton-level contribution is usually infrared-divergent, and only the sum of all contributions produces a finite and physically well-defined result. To enable the implementation of the individual contributions, one typically introduces a subtraction method to separate finite and divergent parts and to cancel divergences among different contributions.

Perturbative calculations at next-to-leading order (NLO) in QCD, based on parton-level event generators, are available for a very broad spectrum of processes, often ranging to very high final state multiplicities. Substantial efforts are currently under way to extend calculations for low-multiplicity benchmark processes to next-to-next-to-leading order

(NNLO) in QCD. A pioneering calculation in this context was the NNLO corrections to event shapes and jet rates in electron-positron annihilation [1–3], based on the antenna subtraction method [4] for the handling of infrared-divergent kinematic contributions.

In this paper, we describe the program **EERAD3**, which we originally developed in the context of [1, 2, 5]. Compared to the original version, we have made substantial improvements in efficiency and usability. The program structure being very modular, it allows the user to either run the program for the production of jet rates and event shape distributions that are already implemented, or to extend the program towards new sets of observables. The program code can be downloaded at <http://eerad3.hepforge.org>.

## 2. Event shapes and jet cross sections in perturbative QCD

Event shape observables have proven very useful to characterise hadronic final states in electron-positron annihilation without the need to define jets. These observables can be divided into classes, according to the minimal number of final-state particles required for them to be non-vanishing: the most common variables require three particles (and are thus closely related to three-jet final states), while some other variables require at least four final state particles.

Among the event shapes requiring three-particle final states, six “classical” event shape variables can be calculated directly with the program **EERAD3**. These are the thrust  $T$  [6], the normalised heavy jet mass  $M_H^2/s$  [7], the wide and total jet broadenings  $B_W$  and  $B_T$  [8], the  $C$ -parameter [9] and the transition from three-jet to two-jet final states in the Durham jet algorithm  $y_{23}$  [10]. The definitions of these variables are collected and described in more detail in Ref. [1].

The perturbative expansion for the distribution of an event shape observable  $y$  up to NNLO at the centre-of-mass energy  $\sqrt{s}$  and renormalisation scale  $\mu^2 = s$ , with  $\alpha_s \equiv \alpha_s(\sqrt{s})$ , is given by

$$\frac{1}{\sigma_{\text{had}}} \frac{d\sigma}{dy} = \left(\frac{\alpha_s}{2\pi}\right) \frac{d\bar{A}}{dy} + \left(\frac{\alpha_s}{2\pi}\right)^2 \frac{d\bar{B}}{dy} + \left(\frac{\alpha_s}{2\pi}\right)^3 \frac{d\bar{C}}{dy} + \mathcal{O}(\alpha_s^4). \quad (2.1)$$

Here the event shape distribution is normalised to the total hadronic cross section  $\sigma_{\text{had}}$ . The latter can be expanded as

$$\sigma_{\text{had}} = \sigma_0 \left( 1 + \frac{3}{2} C_F \left( \frac{\alpha_s}{2\pi} \right) + K_2 \left( \frac{\alpha_s}{2\pi} \right)^2 + \mathcal{O}(\alpha_s^3) \right), \quad (2.2)$$

where the Born cross section for  $e^+e^- \rightarrow q\bar{q}$  is

$$\sigma_0 = \frac{4\pi\alpha}{3s} N e_q^2, \quad (2.3)$$

assuming massless quarks. The constant  $K_2$  is given by [11],

$$K_2 = \frac{1}{4} \left[ -\frac{3}{2} C_F^2 + C_F C_A \left( \frac{123}{2} - 44\zeta_3 \right) + C_F T_R N_F (-22 + 16\zeta_3) \right], \quad (2.4)$$

with  $C_A = N$ ,  $C_F = (N^2 - 1)/(2N)$ ,  $T_R = 1/2$ , and  $N_F$  light quark flavours.

The program **EERAD3** computes the perturbative coefficients  $A$ ,  $B$  and  $C$ , which are normalised to  $\sigma_0$ :

$$\frac{1}{\sigma_0} \frac{d\sigma}{dy} = \left(\frac{\alpha_s}{2\pi}\right) \frac{dA}{dy} + \left(\frac{\alpha_s}{2\pi}\right)^2 \frac{dB}{dy} + \left(\frac{\alpha_s}{2\pi}\right)^3 \frac{dC}{dy} + \mathcal{O}(\alpha_s^4). \quad (2.5)$$

$A$ ,  $B$  and  $C$  are straightforwardly related to  $\bar{A}$ ,  $\bar{B}$  and  $\bar{C}$ :

$$\bar{A} = A, \quad \bar{B} = B - \frac{3}{2}C_F A, \quad \bar{C} = C - \frac{3}{2}C_F B + \left(\frac{9}{4}C_F^2 - K_2\right) A. \quad (2.6)$$

As these coefficients are computed at a renormalisation scale fixed to the centre-of-mass energy, they depend only on the value of the observable  $y$ . Electroweak corrections, calculated in [12,13], are not included in the present version of **EERAD3**. Further, the pure-singlet contribution from three-gluon final states to three-jet observables was found to be negligible [14] and is discarded.

The QCD coupling constant evolves according to the renormalisation group equation, which reads to NNLO:

$$\mu^2 \frac{d\alpha_s(\mu)}{d\mu^2} = -\alpha_s(\mu) \left[ \beta_0 \left(\frac{\alpha_s(\mu)}{2\pi}\right) + \beta_1 \left(\frac{\alpha_s(\mu)}{2\pi}\right)^2 + \beta_2 \left(\frac{\alpha_s(\mu)}{2\pi}\right)^3 + \mathcal{O}(\alpha_s^4) \right] \quad (2.7)$$

with the following coefficients in the  $\overline{\text{MS}}$ -scheme:

$$\begin{aligned} \beta_0 &= \frac{11C_A - 4T_R N_F}{6}, \\ \beta_1 &= \frac{17C_A^2 - 10C_A T_R N_F - 6C_F T_R N_F}{6}, \\ \beta_2 &= \frac{1}{432} (2857C_A^3 + 108C_F^2 T_R N_F - 1230C_F C_A T_R N_F - 2830C_A^2 T_R N_F \\ &\quad + 264C_F T_R^2 N_F^2 + 316C_A T_R^2 N_F^2). \end{aligned} \quad (2.8)$$

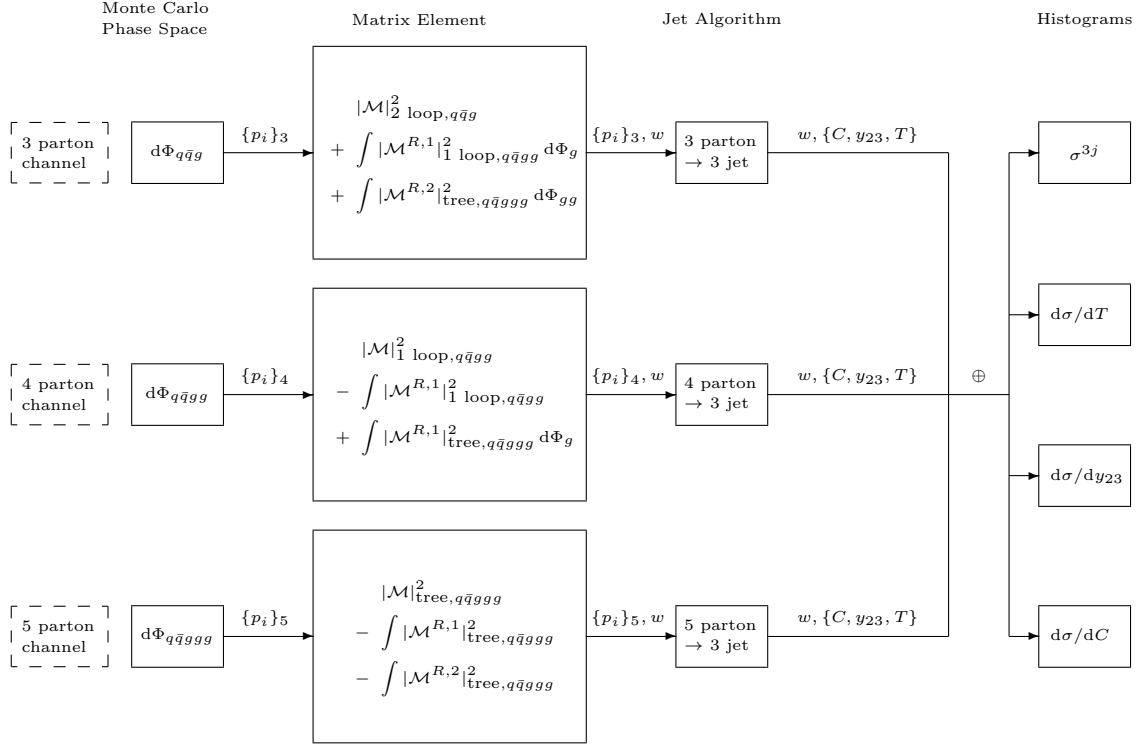
Equation (2.7) is solved by introducing  $\Lambda$  as integration constant with  $L = \log(\mu^2/\Lambda^2)$ , yielding the running coupling constant:

$$\alpha_s(\mu) = \frac{2\pi}{\beta_0 L} \left( 1 - \frac{\beta_1}{\beta_0^2} \frac{\log L}{L} + \frac{1}{\beta_0^2 L^2} \left( \frac{\beta_1^2}{\beta_0^2} (\log^2 L - \log L - 1) + \frac{\beta_2}{\beta_0} \right) \right). \quad (2.9)$$

In terms of the running coupling  $\alpha_s(\mu)$ , the NNLO (non-singlet) expression for event shape distributions therefore becomes

$$\begin{aligned} \frac{1}{\sigma_{\text{had}}} \frac{d\sigma}{dy}(s, \mu^2, y) &= \left(\frac{\alpha_s(\mu)}{2\pi}\right) \frac{d\bar{A}}{dy} + \left(\frac{\alpha_s(\mu)}{2\pi}\right)^2 \left( \frac{d\bar{B}}{dy} + \frac{d\bar{A}}{dy} \beta_0 \log \frac{\mu^2}{s} \right) \\ &\quad + \left(\frac{\alpha_s(\mu)}{2\pi}\right)^3 \left( \frac{d\bar{C}}{dy} + 2 \frac{d\bar{B}}{dy} \beta_0 \log \frac{\mu^2}{s} + \frac{d\bar{A}}{dy} \left( \beta_0^2 \log^2 \frac{\mu^2}{s} + \beta_1 \log \frac{\mu^2}{s} \right) \right) \\ &\quad + \mathcal{O}(\alpha_s^4). \end{aligned} \quad (2.10)$$

We emphasise again that the program **EERAD3** computes the perturbative coefficients  $A$ ,  $B$  and  $C$  defined in eq. (2.5), where the renormalisation scale has been fixed to the centre-of-mass energy. However, for convenience of the user, we include an auxiliary program



**Figure 1:** Schematic structure of **EERAD3**. The event shape observables  $C, y_{23}, T$  are for illustration, other observables can be calculated as well.

`eerad3.dist.f` in the program package, which produces results according to eq. (2.10) and performs scale variations. More details about the usage of `eerad3.dist.f` are given in Section 4.3.

The program **EERAD3** has been used to compute event-shape distributions [1] and their moments [15] as well as jet rates [2]. An independent validation of these results (also using the antenna subtraction method), resolving problems with large-angle soft terms [16], was made in [3, 17–19]. These results enabled a substantial number of precision QCD studies based on the reanalysis of LEP data [20–30]. In the dijet-limit, resummation techniques predict the logarithmically enhanced terms in the event shape distributions. These were derived independently [30–33] and served as validation of the **EERAD3** results.

### 3. Structure of the program **EERAD3**

The program **EERAD3** computes the three-parton, four-parton and five-parton contributions to hadronic final states in  $e^+e^-$  annihilation, and combines them with subtraction terms appropriate to NNLO in QCD. Any infrared-safe quantity derived from three-particle final states can thus be computed with this program to  $\mathcal{O}(\alpha_s^3)$ . The structure of the program is depicted in Figure 1.

The source code of **EERAD3** consists of the following files:

`eerad3.f`: the main program

<code>3jme.f:</code>	two-loop three-parton matrix elements, from [34]
<code>aversub0.f:</code>	subroutines for subtraction at NLO
<code>aversub1.f:</code>	subroutines for subtraction of double real radiation at NNLO
<code>aversub2.f:</code>	subroutines for subtraction of one-loop real radiation at NNLO
<code>brem.f:</code>	real radiation matrix elements
<code>ecuts.f:</code>	jet algorithms and event shape definitions
<code>eerad3lib.f:</code>	library with special functions and auxiliary routines
<code>histo.f:</code>	histogram handling routines
<code>hplog.f:</code>	one-dimensional harmonic polylogarithms, from [35]
<code>phaseee.f:</code>	phase space routines
<code>sig.f:</code>	parton-level cross sections and subtraction terms
<code>tdhpl.f:</code>	two-dimensional harmonic polylogarithms, from [36]
<code>virt.f:</code>	one-loop four-parton matrix elements, from [37]

Their content and function is outlined in the following. Two auxiliary programs are `eerad3.combine.f` and `eerad3.dist.f`.

### 3.1 Main program

The main program `eerad3.f` is steering the input/output and the Monte Carlo integration. The subroutine `readinit` reads the input file `eerad3.input` and initialises the settings for the colour factors and observables to be calculated, as well as for the Monte Carlo integration. The calculation of the different parts contributing to the NNLO cross section is performed through a call to the subroutine `cross`. In this subroutine, the Monte Carlo integration is done with `vegas` [38], proceeding in two steps: first a grid is constructed, then the actual integration is performed based on this grid. The grid step can also be skipped once a grid is produced and saved to a file by setting `iwarm=0` in `eerad3.input`. Likewise, `iprod=0` skips the integration and production of the histograms. The `vegas` integration routines, adapted for the program `EERAD3`, are defined in the file `eerad3lib.f`.

### 3.2 Cross section

The cross section consists of three basic parts which correspond to different particle multiplicities in the final state and therefore are integrated separately.

- **sig3** corresponds to the leading order kinematics with three identified particles in the final state. This part contains the Born cross section as well as the one-loop and two-loop virtual corrections (from `3jme.f`, using `hplog.f` and `tdhpl.f`) and the finite remainders of the integrated subtraction terms cancelling the poles of the two-loop virtual corrections.
- **sig4** corresponds to four particle kinematics at tree-level or one loop (from `virt.f`), where one of the final state particles can be theoretically unresolved (soft or collinear), and therefore requires the inclusion of infrared subtraction terms.

- **sig5** corresponds to five particle kinematics (from **brem.f**), where two of the final state particles can be theoretically unresolved, requiring the inclusion of infrared subtraction terms for unresolved double real radiation.

This basic structure is reflected by the functions **sig3a**, **sig4a**, **sig5a** defined in the file **sig.f**. We documented the expressions implemented for matrix elements and subtraction terms in these functions in [5].

The subtraction terms are contained in the files **aversub\*.f**, where **aversub0.f** contains the antenna functions and momentum mappings for the singly unresolved parts, while **aversub1.f** contains the functions and momentum mappings for the doubly unresolved subtraction terms and **aversub2.f** for the single unresolved subtraction terms at one loop. The one-loop and two-loop matrix elements [34] are expressed in terms of one-dimensional [39] and two-dimensional [40] harmonic polylogarithms. These functions are evaluated in **hplog.f** [35] and **tdhpl.f** [36].

### 3.3 Phase space integration

Based on the **vegas** integration variables, the routines in **phaseeee.f** generate phase space points for three, four and five partons. The four-parton and five-parton phase spaces are decomposed into wedges (6 wedges at the four-parton level and 45 wedges at the five-parton level). Each wedge contains only certain classes of unresolved limits, such that a phase space parametrisation appropriate to these limits can be used. At the four-parton level, permutations of a single parametrisation are sufficient, while two parametrisations (double single collinear (two pairs of collinear partons) and triple collinear (three collinear partons)) are used at the five-parton level. An angular rotation of unresolved pairs of momenta is performed inside each wedge, such that angular correlations [41–43] in the collinear splitting functions can be cancelled out by averaging. The four-parton phase space contains at most single collinear limits, such that two angular-correlated phase space points are sufficient. Four angular-correlated phase space points are generated for the five-parton phase space to account for angular terms in double single collinear and triple collinear limits.

### 3.4 Event shapes and jet cross sections

Based on the four-momenta provided by the phase space generator (or the phase space mappings), parton-level values for event shapes and jet transition parameters are computed in **ecuts.f**. To add further shape variables or different jet algorithms, the user can extend these routines. The parton momenta are contained in **ppar(i,j)**, where **i=(1:4)** enumerates the momentum components (with energy in the fourth entry) and **j=(1:5)** enumerates the momenta. For each event shape variable or jet transition rate, infrared-safe implementations for **npar=3,4,5**, the number of partons in the final state, must be provided.

In **ecuts.f**, events are accepted if they pass the cuts for at least one of the distributions that are computed. Different distributions can be computed simultaneously, such that the

result of the vegas integration will usually not have a physical meaning (except if the program is run for a single observable, potentially selecting a specific moment weight). Instead, the physical distributions are contained in the histograms generated by the program.

### 3.5 Calculation of moments of event shape observables

The  $n$ -th moment of an event shape observable  $y$  is defined as

$$\langle y^n \rangle = \frac{1}{\sigma_{\text{had}}} \int_0^{y_{\text{max}}} y^n \frac{d\sigma}{dy} dy, \quad (3.1)$$

where  $y_{\text{max}}$  is the kinematically allowed upper limit of the observable. As the calculation of moments involves an integration over the full phase space, they offer a way of studying an observable which is complementary to the use of distributions; in particular, they are useful to investigate non-perturbative effects. Recent studies of event shape moments can be found in [15, 18, 21, 26, 28, 44].

The perturbative QCD expansion of  $\langle y^n \rangle$  is given by [15]

$$\langle y^n \rangle(s, \mu^2 = s) = \left( \frac{\alpha_s}{2\pi} \right) \bar{\mathcal{A}}_{y,n} + \left( \frac{\alpha_s}{2\pi} \right)^2 \bar{\mathcal{B}}_{y,n} + \left( \frac{\alpha_s}{2\pi} \right)^3 \bar{\mathcal{C}}_{y,n} + \mathcal{O}(\alpha_s)^4. \quad (3.2)$$

Just as in eq. (2.10), the NNLO expression for an event shape moment measured at centre-of-mass energy squared  $s$  evaluated at renormalisation scale  $\mu$  becomes,

$$\begin{aligned} \langle y^n \rangle(s, \mu^2) = & \left( \frac{\alpha_s(\mu)}{2\pi} \right) \bar{\mathcal{A}}_{y,n} + \left( \frac{\alpha_s(\mu)}{2\pi} \right)^2 \left( \bar{\mathcal{B}}_{y,n} + \bar{\mathcal{A}}_{y,n} \beta_0 \log \frac{\mu^2}{s} \right) \\ & + \left( \frac{\alpha_s(\mu)}{2\pi} \right)^3 \left( \bar{\mathcal{C}}_{y,n} + 2\bar{\mathcal{B}}_{y,n} \beta_0 \log \frac{\mu^2}{s} + \bar{\mathcal{A}}_{y,n} \left( \beta_0^2 \log^2 \frac{\mu^2}{s} + \beta_1 \log \frac{\mu^2}{s} \right) \right) \\ & + \mathcal{O}(\alpha_s^4). \end{aligned} \quad (3.3)$$

To calculate a particular moment for a particular event shape variable (`iaver=1..7`), **EERAD3** must be run with `imom` set to the desired moment number. The result is then obtained as the integration output. Note that the default setting `imom=1` should be used to calculate distributions and jet cross sections.

### 3.6 Booking of results into histograms

The histograms are defined in `histo.f`, using a generic, observable-independent histogram manager. The latter is defined in the subroutine `ghiman` in `eerad3lib.f`. The communication with the histogram manager is made through a subroutine `bino`, which can be extended by the user. For each histogram, four calls to the histogram manager are specified in `bino`; `histoi` initialises the histogram, `histoa` adds an event to the histogram, `histoe` computes the histogram errors and `histow` writes out the final histograms. Each histogram is identified by its identifier number `histoID`, its minimum and maximum bin values `bmin/bmax` and its number of bins `nbins`. To change histogram bins and boundaries, or to introduce histograms for new variables, the user can modify the appropriate calls in `bino`.



The EERAD3 program produces histograms with linear or logarithmic binning. Since the phase space requirements and variable ranges differ substantially for the different binnings, they can not be produced in a single run. For  $0 \leq \text{iaver} \leq 5$ , only the linearly binned histograms are computed, while  $6 \leq \text{iaver} \leq 8$  produces only histograms with logarithmic binning.

For each event shape variable, three types of histograms can be produced: in the case of the thrust distribution in  $\tau = 1 - T$ , these are for example

1.  $\langle \tau \rangle$  distribution  $\frac{\tau}{\sigma} \frac{d\sigma}{d\tau}$  for four different binnings, **nbin**=200,100,50,25, with **bmin**=0, **bmax**=0.5. Produced for **iaver**=0 or **iaver**=4.
2.  $\tau$  distribution  $\frac{1}{\sigma} \frac{d\sigma}{d\tau}$  for four different binnings, **nbin**=200,100,50,25, with **bmin**=0, **bmax**=0.5. Produced for **iaver**=0 or **iaver**=4.
3.  $-\ln(\tau)$  distribution  $\frac{1}{\sigma} \frac{d\sigma}{d(-\ln(\tau))}$  for three different binnings, **nbin**=100,50,25, with **bmin**=0, **bmax**=10. Produced for **iaver**=8.

The  $C$ -parameter has different default linear binnings, **nbin**=400,200,100,50.

The jet rates and transition parameters are always booked into logarithmic histograms, which means that the produced histograms will be  $R_3(-\ln(y_{\text{cut},D})) = \frac{1}{\sigma} \sigma_3(-\ln(y_{\text{cut},D}))$  for the  $y_{\text{cut}}$  dependence of the three-jet rate with the Durham  $k_t$  jet algorithm [10],  $\frac{1}{\sigma} \frac{d\sigma}{d(-\ln(y_{23,D}))}$ , for the 2-to-3-jet transition parameter with the Durham  $k_t$  jet algorithm, analogous for the 3-to-4-jet and 4-to-5-jet transition parameters  $y_{34,D}$  and  $y_{45,D}$ . These histograms are produced using **iaver**=6, while **iaver**=7 produces the same distributions with the Jade jet algorithm [45]. For **iaver**=8 the output consists of all jet distributions based on the Durham  $k_t$  jet algorithm and all logarithmic event shape distributions.

To add additional histograms into **bino**, the user should initialise the histograms with a call to **histoi(histoID,bmin,bmax,nbins)**, and fill the histogram with the corresponding variable value **val** and Monte Carlo weight **wgt** with a call to **histoa(histoID,val,wgt)**. The subroutine **histoe(histoID)** calculates the statistical errors after each **vegas** iteration, and routine **histow(histoID)** writes the histogram to the screen, output to a file with unit number **lun** is done with **histowf(histoID,lun)**. All these subroutines are defined in **eerad3lib.f**.

If the user would like to modify or make additions to the default observable definitions or jet algorithms, the file **ecuts.f** should be edited. The jets are defined in the subroutine **getjet**. The Durham  $k_t$  and Jade algorithms are implemented with different options for the recombination schemes. The default for the Durham  $k_t$  algorithm is the so-called “E-scheme” [45], where the sum of two momentum vectors is defined by the usual vector sum, treating the energy and spatial components on the same footing. The default for the Jade algorithm is the “E0-scheme” [45], where only the energy components are simply summed, while the sum of the spatial components is weighted by a factor. The latter is given by the sum of the energy components divided by the modulus of the sum of the spatial components. The different recombination schemes are defined in the subroutine **jetco**, which is called by **getjet**. All these subroutines, as well as the definitions of the event shape observables, are defined in the file **ecuts.f**.

The cuts which are read from the input card are set in the subroutine `readinit` in the main program `eerad3.f` and shared with the subroutine `ecuts` in `ecuts.f`.

The file `eerad3.combine.f` allows to combine results from several different runs into a single set of histograms as described in section 4.2. While `EERAD3` computes only the perturbative coefficients  $A, B, C$ , the program `eerad3_dist.f` can be used to construct cross sections and distributions according to (2.10), and to perform scale variations. Detailed instructions for the usage of `eerad3.combine.f` and `eerad3_dist.f` are given in Sections 4.2 and 4.3, respectively.

## 4. Usage of EERAD3

### 4.1 Main program eerad3

The main program `EERAD3` computes individual colour structures of the perturbative coefficients of event shapes and jet rates. It is controlled through an input card. The program can be compiled by simply executing the command `make`, which will produce an executable called `eerad3`. The calculation can be started with command line options

```
$ eerad3 -i filename.input -n XX
```

where `filename.input` is the name of the input card file and `XX` is an integer between 0 and 99 selecting the random seed for runs with independent statistics. Default values `filename.input=eerad3.input` and `XX=0` are inserted if no command line options are given.

### Structure of the input card

A typical example for an input card looks as follows:

```
1d-5      ! y0
0          ! iaver
0.0025    ! cutvar
1          ! imom
1          ! iang
-2         ! nloop
1          ! icol
Z          ! ichar
1 1        ! iwarm iprod
5 5        ! itmax1 itmax2
5000 30000 200000 ! nshot3 nshot4 nshot5
```

The individual entries are:

**y0:** technical cut-off for the phase space integration (dimensionless), should be between  $10^{-5}$  and  $10^{-8}$ .

**iaver:** selects the observables to be computed:  
0: all event shape distributions  $(B_W, C, M_H^2/s, (1 - T), B_T)$ ,

	1: wide jet broadening $B_W$ .
	2: $C$ -parameter $C$ .
	3: heavy jet mass $M_H^2/s$ .
	4: thrust $1 - T$ .
	5: total jet broadening: $B_T$ .
	6: jet rates and transition parameters in the Durham $k_t$ algorithm: $R_3, R_4, R_5, y_{23}, y_{34}, y_{45}$ .
	7: jet rates and transition parameters in the Jade algorithm: $R_3, R_4, R_5, y_{23}, y_{34}, y_{45}$ .
	8: all jet distributions in the Durham $k_t$ algorithm and all logarithmic event shape distributions.
<b>cutvar:</b>	lower cut-off on distributions, should be at least one order of magnitude larger than <b>y0</b> .
<b>imom:</b>	moment number applied as weight: if computing distributions, should be set to <b>imom=1</b>
<b>iang:</b>	angular optimisation of phase space: on ( <b>iang</b> = 1) or off ( <b>iang</b> = 2).
<b>nloop:</b>	perturbative order: LO ( <b>nloop</b> =0), NLO ( <b>nloop</b> =-1), NNLO ( <b>nloop</b> =-2).
<b>icol:</b>	colour factor: 0: sum all colour factors, 1: NLO $N$ , 2: NLO $1/N$ , 3: NLO $N_F$ , 1: NNLO $N^2$ , 2: NNLO $N^0$ , 3: NNLO $1/N^2$ , 4: NNLO $N_F N$ , 5: NNLO $N_F/N$ , 6: NNLO $N_F^2$ .
<b>ichar:</b>	one character to identify output files.
<b>iwarm:</b>	produce a warm up integration grid ( <b>iwarm</b> =1) or read the grid from files ( <b>iwarm</b> =0).
<b>iprod:</b>	produce histograms: yes ( <b>iprod</b> =1) or no ( <b>iprod</b> = 0).
<b>itmax1,2:</b>	number of iterations for warm up and production runs.
<b>nshot3,4,5:</b>	number of vegas points for three-parton, four-parton, five-parton channels.

## Calculation of moments of event shapes

As mentioned already in Section 3.5, the  $n$ -th moment of an event shape observable can be calculated by selecting **imom**= $n$  in the input card. Please note that for the calculation of the moments the options **iaver**=0 and **iaver**=8 can not be used. These options calculate a whole set of observables, using an unphysical integrand for the **vegas** integration, which is appropriately re-weighted only for the booking of events into different histograms.

## Structure and naming conventions of the output files

The column structure of the output histogram files is as follows:

variable	observable	error
----------	------------	-------

The filenames are composed as

E[aa].y[bbb].i[c][d].[e][f][g]

with

- [aa]: two-digit identifier for random seed (input from command line)
- [bbb]: value of  $y_0$  in format  $\text{ndi} = n \cdot 10^i$ .
- [c]: one-character filename identifier  $i_{\text{char}}$ .
- [d]: one-digit colour factor identifier  $i_{\text{col}}$ .
- [e]: one-character identifier of observable selected by  $i_{\text{aver}}$ ,  
where the conventions are:
  - W:  $B_W$
  - C:  $C$
  - M:  $M_H^2/s$
  - T:  $\tau$
  - B:  $B_T$
  - Y: jet transition parameters  $y_{ij}$
  - S: jet rates  $y_n$
- [f]: one-character identifier of distribution type,  
for event shapes:
  - 1: distribution  $1/\sigma_0 y(d\sigma/dy)$
  - 2: distribution  $1/\sigma_0 (d\sigma/dy)$
  - L: distribution  $1/\sigma_0 (d\sigma/dL)$  with  $L = \ln y$for jet rates and transition parameters (logarithmic binning):
  - 3: three-jet rate and  $y_{23}$  transition parameter
  - 4: four-jet rate and  $y_{34}$  transition parameter
  - 5: five-jet rate and  $y_{45}$  transition parameter
- [g]: one-character identifier of bin size (resolution), runs from **a** to **d**,  
corresponding to the binnings described in Section 3.6.

## 4.2 eerad3\_combine

To obtain statistically independent samples, **eerad3** can be run with different random seeds. The combination of different runs is performed by the program **eerad3\_combine**, which reads an input card

```
$ eerad3_combine -i filename.input
```

The default input filename is **eerad3\_combine.input**. The content of this file is as follows

```

0      ! iaver
y1d5   ! frooty
iZ3    ! frooti
tx     ! filetag
1 20   ! minfile maxfile
3      ! nvoid
2      ! ivalid(1)
6      ! ivalid(2)
14     ! ivalid(3)

```

Where:

```

iaver:      selects observables to be computed, as above.
frooty:     four-character filename extension to indicate the value of y0,
            i.e. y[bbb] described above in the filename composition.
frooti:     three-character filename extension to indicate identifier and colour factor.
            i.e. i[c] [d] described above in the filename composition.
filetag:    two-character identifier for combination files.
minfile, maxfile: random seed identifiers to fix range of files to be combined.
nvoid:      number of (void or unused) runs to be excluded from the combination.
ivalid(1:nvoid) random seed identifiers of files to be excluded.

```

The syntax of the combination filenames is as for the individual runs, with the random seed identifier number replaced by `filetag`.

### 4.3 eerad3\_dist

The main program `eerad3` computes the perturbative coefficients of event shape distributions and jet cross sections, as defined in (2.5). Cross sections and distributions according to eq. (2.10) are obtained from these coefficients using `eerad3_dist`, which read an input card

```
$ eerad3_dist -i filename.input
```

The default input filename is `eerad3_dist.input`. The content of this file is as follows

```

8      ! iaver
E00.y1d8.iL0   !   L0
Etx.y1d8.iN1   !  NLO icol=1
Etx.y1d8.iN2   !  NLO icol=2
Etx.y1d8.iN3   !  NLO icol=3
Etx.y1d6.iZ1   ! NNLO icol=1
Etx.y1d6.iZ2   ! NNLO icol=2
Etx.y1d6.iZ3   ! NNLO icol=3
Etx.y1d6.iZ4   ! NNLO icol=4
Etx.y1d6.iZ5   ! NNLO icol=5

```

```

Etx.y1d6.iZ6      ! NNLO icol=6
EEt.091.1189      ! outfile
0.1189d0          ! alphas(MZ)
91.1889d0         ! MZ
91.1889d0         ! roots
1.d0              ! xmu

```

Where:

```

iaver:           selects observables to be computed, as above.
L0,...:          names (without extensions) of the files containing the histograms;
                  fixed to 12 characters by construction.
outfile:         name (without extensions) for output files.
alphas(MZ):      value of the strong coupling constant  $\alpha_s(M_Z)$ .
MZ:             mass of  $Z$ -boson.
roots:           $e^+e^-$  centre-of-mass energy  $\sqrt{s}$ .
xmu:            default value of renormalisation scale:  $\mu = x_\mu \sqrt{s}$ .

```

The following files are produced:

```

outfile.xNNLO.[e][f][g]:      distributions up to NNLO.
outfile.muddep.[e][f][g]:     distributions with range of scale variation.
outfile.muran.[e][f][g]:     percentages of scale variation.
outfile.histA.[e][f][g]:     perturbative coefficient  $\bar{A}$ .
outfile.histB.[e][f][g]:     perturbative coefficient  $\bar{B}$ .
outfile.histC.[e][f][g]:     perturbative coefficient  $\bar{C}$ .

```

The column structure of the xNNLO files is as follows:

```

variable      sigmaL0      sigmaNNLO      sigmaNNLO      errNNLO

```

In here **errNNLO** is the numerical integration error at NNLO.

The scale variation uncertainty is determined by finding the minimum and maximum of the cross section over a range  $\mu/2$  to  $2\mu$  around the default value  $\mu$ . To resolve extrema inside this interval, 20 logarithmically spaced points are computed. The results are contained in the **muddep** files, which have the following structure:

```

variable  sigL0  errL0  sigNLO  errNLO  sigNNLO  errNNLO

```

The errors here are the errors from scale variations.

To quantify the precision of the predictions, scale uncertainties in per cent are provided in the **muran** files, with the conventions:

```

variable  %(L0)      %(NLO)      %(NNLO)

```

The scale variation errors in the **muddep** and **muran** files do not take proper account of the integration errors on the NNLO coefficients. For a full consideration of error propagation, it is therefore recommended to produce xNNLO files with different values of **xmu**.

The coefficients  $\bar{A}$ ,  $\bar{B}$  and  $\bar{C}$  are defined in (2.6), the structure of the **histoA**, **histoB** and **histoC** files is as described in Section 4.1 above.

## 5. Summary

The program **EERAD3** computes jet cross sections and event shapes in electron-positron annihilation to order  $\alpha_s^3$ , corresponding to NNLO accuracy in perturbative QCD. The standard set of event shape observables and jet rates studied at LEP is implemented in the program. We have documented the structure and usage of **EERAD3** in detail, and explained how the user can extend the program to compute other shape variables or use alternative jet algorithms. The program is available at <http://eerad3.hepforge.org>.

## Acknowledgements

GH would like to thank the Physics Institute at the University of Zurich for hospitality while parts of this project were carried out. This research was supported in part by the UK Science and Technology Facilities Council, in part by the Swiss National Science Foundation (SNF) under contracts PP00P2-139192 and 200020-149517 and in part by the European Commission through the “LHCPhenoNet” Initial Training Network PITN-GA-2010-264564. EWNG gratefully acknowledges the support of the Wolfson Foundation, the Royal Society and the Pauli Center for Theoretical Studies.

## References

- [1] A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover and G. Heinrich, JHEP **0712** (2007) 094 [arXiv:0711.4711].
- [2] A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover and G. Heinrich, Phys. Rev. Lett. **100** (2008) 172001 [arXiv:0802.0813].
- [3] S. Weinzierl, JHEP **0906** (2009) 041 [arXiv:0904.1077].
- [4] A. Gehrmann-De Ridder, T. Gehrmann and E.W.N. Glover, JHEP **0509** (2005) 056 [hep-ph/0505111].
- [5] A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover and G. Heinrich, JHEP **0711** (2007) 058 [arXiv:0710.0346].
- [6] S. Brandt, C. Peyrou, R. Sosnowski and A. Wroblewski, Phys. Lett. **12** (1964) 57; E. Farhi, Phys. Rev. Lett. **39** (1977) 1587.
- [7] L. Clavelli and D. Wyler, Phys. Lett. B **103** (1981) 383.
- [8] P.E.L. Rakow and B.R. Webber, Nucl. Phys. B **191** (1981) 63;  
S. Catani, G. Turnock and B. R. Webber, Phys. Lett. B **295** (1992) 269.
- [9] G. Parisi, Phys. Lett. B **74** (1978) 65;  
J.F. Donoghue, F.E. Low and S.Y. Pi, Phys. Rev. D **20** (1979) 2759.
- [10] S. Catani, Y.L. Dokshitzer, M. Olsson, G. Turnock and B.R. Webber, Phys. Lett. B **269** (1991) 432;  
N. Brown and W.J. Stirling, Phys. Lett. B **252** (1990) 657; Z. Phys. C **53** (1992) 629;  
W.J. Stirling *et al.*, Proceedings of the Durham Workshop, J. Phys. **G17** (1991) 1567.
- [11] K.G. Chetyrkin, J.H. Kühn and A. Kwiatkowski, Phys. Rept. **277** (1996) 189.

- [12] A. Denner, S. Dittmaier, T. Gehrmann and C. Kurz, Phys. Lett. B **679** (2009) 219 [arXiv:0906.0372].
- [13] A. Denner, S. Dittmaier, T. Gehrmann and C. Kurz, Nucl. Phys. B **836** (2010) 37 [arXiv:1003.0986].
- [14] J.J. van der Bij and E.W.N. Glover, Nucl. Phys. B **313** (1989) 237.
- [15] A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover and G. Heinrich, JHEP **0905** (2009) 106 [arXiv:0903.4658].
- [16] S. Weinzierl, Phys. Rev. Lett. **101** (2008) 162001 [arXiv:0807.3241].
- [17] S. Weinzierl, JHEP **0907** (2009) 009 [arXiv:0904.1145].
- [18] S. Weinzierl, Phys. Rev. D **80** (2009) 094018 [arXiv:0909.5056].
- [19] S. Weinzierl, Eur. Phys. J. C **71** (2011) 1565 [Erratum-ibid. C **71** (2011) 1717] [arXiv:1011.6247].
- [20] S. Bethke, S. Kluth, C. Pahl and J. Schieck [JADE Collaboration], Eur. Phys. J. C **64** (2009) 351 [arXiv:0810.1389].
- [21] C. Pahl, S. Bethke, O. Biebel, S. Kluth and J. Schieck, Eur. Phys. J. C **64** (2009) 533 [arXiv:0904.0786].
- [22] J. Schieck *et al.* [JADE Collaboration], Eur. Phys. J. C **73** (2013) 2332 [arXiv:1205.3714].
- [23] G. Dissertori, A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover, G. Heinrich, G. Luisoni and H. Stenzel, JHEP **0908** (2009) 036 [arXiv:0906.3436].
- [24] G. Dissertori, A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover, G. Heinrich and H. Stenzel, Phys. Rev. Lett. **104** (2010) 072002 [arXiv:0910.4283].
- [25] G. Abbiendi *et al.* [OPAL Collaboration], Eur. Phys. J. C **71** (2011) 1733 [arXiv:1101.1470].
- [26] T. Gehrmann, M. Jaquier and G. Luisoni, Eur. Phys. J. C **67** (2010) 57 [arXiv:0911.2422].
- [27] R. Abbate, M. Fickinger, A. H. Hoang, V. Mateu and I. W. Stewart, Phys. Rev. D **83** (2011) 074021 [arXiv:1006.3080].
- [28] R. Abbate, M. Fickinger, A. H. Hoang, V. Mateu and I. W. Stewart, Phys. Rev. D **86** (2012) 094002 [arXiv:1204.5746].
- [29] T. Gehrmann, G. Luisoni and P. F. Monni, Eur. Phys. J. C **73** (2013) 2265 [arXiv:1210.6945].
- [30] T. Becher and M. D. Schwartz, JHEP **0807** (2008) 034 [arXiv:0803.0342].
- [31] Y. -T. Chien and M. D. Schwartz, JHEP **1008** (2010) 058 [arXiv:1005.1644].
- [32] P. F. Monni, T. Gehrmann and G. Luisoni, JHEP **1108** (2011) 010 [arXiv:1105.4560].
- [33] T. Becher and G. Bell, JHEP **1211** (2012) 126 [arXiv:1210.0580].
- [34] L.W. Garland, T. Gehrmann, E.W.N. Glover, A. Koukoutsakis and E. Remiddi, Nucl. Phys. B **627** (2002) 107 [hep-ph/0112081] and **642** (2002) 227 [hep-ph/0206067].
- [35] T. Gehrmann and E. Remiddi, Comput. Phys. Commun. **141** (2001) 296 [hep-ph/0107173].
- [36] T. Gehrmann and E. Remiddi, Comput. Phys. Commun. **144** (2002) 200 [hep-ph/0111255].



- [37] E.W.N. Glover and D.J. Miller, Phys. Lett. B **396** (1997) 257 [hep-ph/9609474];  
J.M. Campbell, E.W.N. Glover and D.J. Miller, Phys. Lett. B **409** (1997) 503  
[hep-ph/9706297];  
Z. Bern, L.J. Dixon and D.A. Kosower, Nucl. Phys. B **513** (1998) 3 [hep-ph/9708239].
- [38] G.P. Lepage, J. Comput. Phys. **27** (1978) 192.
- [39] E. Remiddi and J.A.M. Vermaseren, Int. J. Mod. Phys. A **15** (2000) 725 [hep-ph/9905237].
- [40] T. Gehrmann and E. Remiddi, Nucl. Phys. B **601** (2001) 248 [hep-ph/0008287].
- [41] S. Weinzierl, Phys. Rev. D **74** (2006) 014020 [hep-ph/0606008].
- [42] S. Catani and M.H. Seymour, Nucl. Phys. B **485** (1997) 291 [hep-ph/9605323].
- [43] S. Catani and M. Grazzini, Nucl. Phys. B **570** (2000) 287 [hep-ph/9908523].
- [44] C. Pahl *et al.* [JADE Collaboration], Eur. Phys. J. C **60** (2009) 181.
- [45] S. Bethke, Z. Kunszt, D. E. Soper and W. J. Stirling, Nucl. Phys. B **370** (1992) 310  
[Erratum-ibid. B **523** (1998) 681].